Creating and Running Circuits for the TM-4

Dave Galloway

Edward S. Rogers Sr. Department of Electrical and Computer Engineering
University of Toronto

March 2005

Compiling a Circuit for the TM-4

Circuits for the TM-4 can be prepared in one of two formats: VHDL (.vhd) format, or Verilog (.v) format. This document will use VHDL in its examples, but you should be able to use a .v file wherever a .vhd file is used.

Create one netlist file for each FPGA that you wish to use. Run the `tm4vhd` command like this:

```
tm4vhd file1.vhd file2.vhd file3.vhd file4.vhd
```

This will run the Altera CAD software, creating a new directory called `tm4` containing the programming files for the TM-4. If you want to use a name other than tm4 for the directory, use the `-o dirname` option.

If you do not wish to use a particular FPGA, replace the corresponding `fileN.vhd` argument with the string `BLANK`.

There are 4 user-programmable FPGAs. The TM-4 master clock is named `tm4_glbclk0`. There are two identical copies of this clock signal named `tm4_glbclk1` and `tm4_glbclk2`, which all go to each of the user FPGAs, but are connected to different clock input pins, and can be fed to independent PLLs inside those FPGAs.

Additional tm4vhd options are documented in the "Advanced Compilation Options" section of this document.

Communicating with your Circuit

The TM-4 ports package can be used to communicate with your circuit. See *The TM-4 Ports Package* for details. If you want to use the ports package with one or more of the fpgas, create a port description file for each fpga *before running tm4vhd*, giving them the same names as the corresponding netlists, but with a .ports extension. You must also instantiate a component in your netlist called `tm4_portmux`, passing it the signals that you want to connect to the ports package.

Communicating with External Hardware

There are two video input interfaces and one VGA output interface connected to fpga 0. There are two Firewire (IEEE-1394) interfaces connected to fpga 1.

There are four 40 pin ribbon cable connectors which can be connected to external hardware. Each I/O connector has 25 signal pins, seven grounds, three 3.3V output pins, four 1.5V output pins and one clock output pin. Two of the connectors are wired to fpga 2, and two are connected to fpga 3. To use the signals in one of those fpgas, name them tm4_exp40(49 downto 0). The wiring of the connectors can be seen on page 27 of the file `schematic_revA.pdf` in the `~tm4/doc` directory.

DDR DRAM

There are two DDR DRAM modules connected to each FPGA. Each module contains approximately 1 Gbyte of memory, arranged as 72 bits x 2 chips x 4 banks x 16384 rows x 1024 columns.

To use the memory, take a look at the example circuit in `~tm4/examples/ddr`.

Using the TM-4

In order to use the TM-4, the `tm4mon` daemon must be running on the machine named `crunch.eecg`. This daemon is normally started when the machine is booted, and will continue running. If the daemon stops for some reason, you can re-start it by signing on to that machine and typing:

```
tm4mon &
```

The `tm4` command can then be used to communicate with the tm4mon daemon. See *The tm4mon Program* document for details.

You can check the status of the TM-4 remotely by running the `tm4status` program. It will create an X-window on your screen that shows the current status of the machine.

The `tm4get` command will attempt to reserve the machine for a specified number of minutes (default 10). If it succeeds, your name will be shown on everyone's tm4status display. When you are finished, run the `tm4release` command so that others may use the machine.

To download and start a circuit in the TM-4, run:

```
tm4run [dirname]
```

By default, tm4run will look for the programming files in the tm4 subdirectory. The optional directory name argument will cause it to look in the specified directory.

Checking Circuit Speed

The tm4vhd script will run the Quartus timing analyzer, leaving the output for each FPGA in the tm4 subdirectory. The `tm4timing` script will display a portion of this output.

To set the frequency of the `tm4_glbclk0` clock signal, run:

```
tm4 set_clk 0 MM000000
```

where MM000000 is the new clock frequency in Hertz. It must be a multiple of 1000000 Hz.

Debugging your Circuit

The `tm4jtag` script will report on the values present on the pins of the FPGAs in your design. Each variable is printed on a single line, along with its value. Variables with more than one pin (ie: buses) are printed in hex.

The `-r` flag will make tm4jtag print the same information in a raw format. It prints one line for each pin used. Each line contains the fpga number, the name of the pin on the FPGA, the bit number in the JTAG frame (fairly useless), the value that the FPGA sees on the pin (0 for a low value, 1 for a high), the state of the output enable for that pin (0 if the pin is being driven by the FPGA, 1 otherwise), the value that the FPGA is attempting to set the pin to (only valid if the FPGA has programmed this pin as an output), and the signal name.

Advanced Compilation Options

The tm4vhd command also takes the following options:

`-chips "N M ..."` will recompile only the designs for the chip numbers that you specify. This option can be used to recompile one or more chips of an already compiled design. You must have compiled the whole design beforehand. You must not modify the number of pins on any of the FPGAs, and may not change the names of any of the pins. For example:

```
tm4vhd chipa.vhd chipb.vhd chipc.vhd chipd.vhd
  < test and modify chipa and chipc, but want to leave the other two alone >
tm4vhd -chips "0 2" chipa.vhd chipc.vhd
```

Implementation Internals

CAD Tools Used

The main commercial tool used to compile circuits for the TM-4 is Altera's Quartus. It is automatically invoked by the tm4vhd script. Look at the `tm4vhd` script in `~tm4/bin` for details.

You can add Quartus compilation options to a file called `designname.qsf` (where designname is the name of your toplevel design file) in your design directory. Check the Quartus documentation for details.

Documentation for Quartus can be found at:

`http://www.altera.com/support/software/sof-quartus.html`

The tm4 Directory

The tm4 directory created by tm4vhd has 6 subdirectories: download, fpga0, fpga1, fpga2, fpga3 and netlist. The download directory contains the programming files for each Stratix chip. The fpgaN directories hold the input and output files from the Quartus software for each chip.

The files in each fpgaN subdirectory will include:

File	Purpose
quartus.out	output from the Quartus tool while compiling the chip
*.tan.rpt	output from Quartus's timing analysis for this chip

The netlist directory contains some intermediate files used to create the circuit netlists.